All You May Need for VQA are Image Captions

Soravit Changpinyo* **Doron Kukliansy***
Idan Szpektor **Xi Chen** **Nan Ding** **Radu Soricut**
Google Research
{schangpi,doronk,szpektor,chillxichen,dingnan,rsoricut}@google.com

Abstract

Visual Question Answering (VQA) has benefited from increasingly sophisticated models, but has not enjoyed the same level of engagement in terms of data creation. In this paper, we propose a method that automatically derives VQA examples at volume, by leveraging the abundance of existing image-caption annotations combined with neural models for textual question generation. We show that the resulting data is of high-quality. VQA models trained on our data improve state-of-the-art zero-shot accuracy by double digits and achieve a level of robustness that lacks in the same model trained on human-annotated VQA data.

1 Introduction

Visual Question Answering (VQA) is a complex multimodal task that, to be successfully modeled and evaluated, requires large amounts of annotations that are not naturally produced by existing business processes, the way translation-pair annotations (Guo et al., 2018) or image alt-text annotations (Sharma et al., 2018) are produced.

At present, a main bottleneck for developing robust VQA systems that are useful for downstream applications, such as for visually-impaired people and in the medical and education domains, appears to be a lack of large image-question-answer training triplets (on the order of millions). Manual annotation of such triplets is costly, time-consuming, and prone to a variety of human biases that are difficult to account for (Yuan, 2021). In addition, the brittleness of VQA systems trained on such manual annotations is well-understood and documented (Agrawal et al., 2018; Kafle and Kanan, 2017).

To address the data limitation, we turn to a potential source for creating VQA examples: image-English caption pairs (Chen et al., 2015; Sharma et al., 2018). Large-scale image caption datasets exist with millions (Changpinyo et al., 2021), several hundreds millions (Radford et al., 2021), or even billions (Jia et al., 2021) of examples. Captions come mostly in the form of declarative sentences, e.g., "*two bears are laying down on the ice*". Yet, the task of converting declarative captions into VQA question/answer pairs is still largely unexplored. It requires automatically inducing candidate answers fitting the VQA task, along with their respective questions based on the caption text (Fig. 1). We note that transforming declarative form to interrogative form plus answer(s) seems crucial, as there exists evidence that a vision-and-language model trained on declarative-language data cannot be successfully adapted or transferred "out-of-the-box" for VQA (Wang et al., 2021).

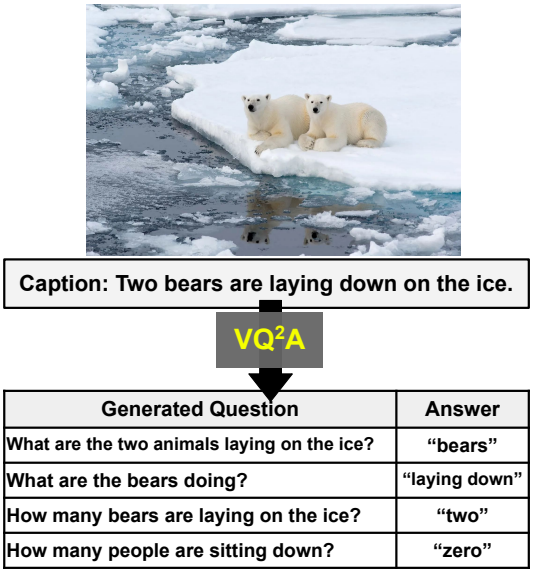

Figure 1: Given an English caption (along with its corresponding image), our VQ^{2}A method generates high-quality question-answer pairs. These image-question-answer triplet data can be automatically produced at volume (millions of examples) and used to effectively train VQA systems.

In this paper, we explore the automatic creation of millions of VQA training data using neural models for textual question generation and question answering. We refer to this method as VQ^{2}A,

* Equal contribution

for **V**isual **Q**uestion Generation with **Q**uestion **A**nswering validation. We demonstrate that VQA models trained on such data, with no exposure to human-annotated VQA data at all, exhibit high zero-shot performance. Our best models obtain 61.1% accuracy on VQA2.0, 52.1% on GQA, around 15-17 points higher than previous zero-shot state-of-the-art results, and getting close to fully-supervised performance. In addition, taking our generated examples as a test set, we provide further evidence for the brittleness of VQA systems built with human-annotated examples, as well as evidence for the robustness of VQA systems built with the automatically-induced $\mathrm{VQ^2A}$ data.

2 Related Work

2.1 Question generation in NLP

Question Generation (QG) is an active research topic in NLP. It is explored as a standalone task (Heilman and Smith, 2009; Nema et al., 2019), as a pre-training task for language models (Narayan et al., 2020) and as a component in solutions for other textual tasks, such as question answering (Alberti et al., 2019; Puri et al., 2020), information retrieval (Mass et al., 2020; Gaur et al., 2021) and generation evaluation (Durmus et al., 2020; Wang et al., 2020; Honovich et al., 2021). There are two main directions to QG: template-based (Heilman and Smith, 2009; Lyu et al., 2021; Dhole and Manning, 2020) and neural-based, with the latter achieving state-of-the-art results (Alberti et al., 2019; Narayan et al., 2020).

2.2 Question generation in computer vision

Question generation in computer vision aims at generating *visual* questions about a given image (or video), either for generating questions without knowing the answer (Mostafazadeh et al., 2016; Zhang et al., 2017; Yang et al., 2018; Uehara et al., 2018; Krishna et al., 2019), e.g., for them to to be answered by humans, or to help improving the VQA task (Kafle et al., 2017; Li et al., 2018; Shah et al., 2019; Xu et al., 2021; Kil et al., 2021; Akula et al., 2021), e.g., for additional evaluation and as means of data augmentation. Such QG models are typically based on VQA triplets as training data, whose language complexity is often limited, or require the collection of visual QG data (Mostafazadeh et al., 2016). We take a different approach by leveraging models trained on *textual* QA datasets instead.

Multiple works leverage image captions or video transcripts as training sources (Ren et al., 2015a; Banerjee et al., 2021; Yang et al., 2021a; Lee et al., 2021). In this approach, question-answer pairs are automatically generated from the text, ignoring the visual source, and are then combined with the related image/video to produce image-question-answer triplets. Banerjee et al. (2021) propose WeaQA, in which they generate questions from MSCOCO image captions (Chen et al., 2015) using an improved template-based approach in COCOQA (Ren et al., 2015a) as well as QA-SRL methods, enhanced by paraphrasing and backtranslation for linguistic variations. Lee et al. (2021) similarly train a VQA model from question-answer pairs derived from MSCOCO Captions but only use noun phrases as candidate answers, focusing on using it to verify generated captions but not on the VQA task itself. Yang et al. (2021a) generate question-answer pairs from instructional video ASR transcripts, which are then coupled with the related video.

In this work, we follow this direction, investigating what requires to generate data with good coverage for the VQA task in the image domain. We show that our neural-based textual question generation approach with captions is much more effective than previous approaches. Further, unlike previous work, we also explore automatically-curated out-of-domain image-text data sources.

2.3 Transfer learning for and in VQA

Existing work also explores the relationship between the image captioning task and the VQA task without question generation (Section 2.2). Fisch et al. (2020) perform image captioning by anticipating visual questions (i.e., using VQA data as additional supervision and post-inference evaluation). Wu et al. (2019) generate question-relevant image captions to aid VQA. Yang et al. (2021b) prompt the GPT-3 (Brown et al., 2020) to answer knowledge-based visual questions based on generated captions and tags and a few VQA examples.

Evidence suggests that image-text pre-training, especially when performed at scale, benefits vision-and-language tasks, including VQA (Lu et al., 2019; Li et al., 2019; Chen et al., 2020; Tan and Bansal, 2019; Su et al., 2020; Lu et al., 2020; Zhou et al., 2020; Li et al., 2020; Zhang et al., 2021; Cho et al., 2021; Wang et al., 2021; Yuan et al., 2021). However, these approaches do not work

well without fine-tuning on the downstream VQA data (Wang et al., 2021). Further, prompt-based learning and inference (Liu et al., 2021) from a pre-trained image-text model that works for VQA is still an open research problem. In contrast, our approach directly works with the training data, explicitly transforms them into the interrogative form of question-answer pairs.

Our focus is the *zero-shot* transfer setting in WeaQA (Banerjee et al., 2021) in which no manually-created VQA triplets are available during training. Note that the term zero-shot here is different from the one used in (Teney and Hengel, 2016), in which the model still has access to manually-created VQA triplets but is evaluated with unseen questions at test time. Similar to this, Chao et al. (2018b) explore cross-dataset VQA but they solely focus on human-annotated data along with approaches to transfer.

3 Textual Question Generation for VQA

We study whether automatically producing VQA annotations from existing image-text resources can alleviate or completely replace the need for manual data annotation. We only focus on English in this paper. To this end, we follow and improve upon some of the recent directions in Section 2.2 on automatic question-answer generation from text.

We start with a given dataset of image-caption pairs D=$\{\mathrm{img}_i, \mathrm{cap}_i\}_{i=1}^{N}$. An important assumption we take is that the information conveyed by the caption is, in the vast majority of cases, present in the image, i.e., captions do not contain an excessive amount of external-world or personal knowledge (e.g., "*my friend at my birthday party*").

For each pair $\{\mathrm{img}_i, \mathrm{cap}_i\}$, an initial set of candidate answers $\{\mathrm{a}_{i,j}\}_{j=1}^{M_i}$ is first automatically derived from cap_i. For each such candidate answer, a question is generated by a neural model $\mathrm{q}_{i,j} = QG(\mathrm{a}_{i,j}, \mathrm{cap}_i)$. Each generated question-answer pair undergoes a validation step, and, if validated, is coupled with the corresponding image img_i to induce a VQA example triplet $\{\mathrm{img}_i, \mathrm{q}_{i,j}, \mathrm{a}_{i,j}\}$.

We refer to this method as VQ^{2}A (**V**isual **Q**uestion Generation with **Q**uestion **A**nswering validation). Figure 2 provides an overview of our approach. We next detail the steps in VQ^{2}A.

3.1 Candidate Answer Extraction

The only prior work on neural question generation from captions we are aware of, Lee et al. (2021), focuses on noun phrases as candidate answers. Yet, these are not enough to cover the answer types included in typical VQA benchmarks such as VQA2.0 (as we will show in Section 5.1), such as boolean, attribute, and verb answers, to name a few, which are required for questions like as "*Is there...*", "*What color...*", "*What is the dog doing*". We present a method that covers all of these answer types.

To extract candidate answers from a given caption, we parse it using spaCy[1] and then extract candidates based on the Part-of-Speech (POS) and dependency parse tree annotations, as follows:

Noun Phrases. We extract all noun phrases annotated by spaCy, including named entities.

POS Spans. We extract sequences that begin with an open-class POS (nouns, verbs, adjectives and adverbs), that end with an open-class POS or an adverbial particle, and that do not contain any other POS in between except closed-class POS for determiners, adpositions and conjunctions.

Parse Tree Spans. We consider all sub-trees that include at least one open-class POS and no more than 3 words altogether. We only extract maximal spans, i.e., not extracting sub-trees that are fully included in other extracted sub-trees.

Boolean. Boolean questions are frequent in VQA benchmarks (Goyal et al., 2017). Yet, '*yes*' and '*no*' are not found in captions, and so cannot be extracted as candidates by extracting text spans from captions. To this end, we also add '*yes*' and '*no*' as candidate answers and generate one question per candidate (see Section 3.2).

How many? 0. Captions do not normally contain mentions of '*zero*' object counts. Hence, marking spans in a caption does not generate questions with the answer '*0*'. Therefore, we randomly sample a generated "*How many?*" question (with a non-zero answer) from a different caption and add it with the answer changed to '*zero*' to the candidate set of the target caption. This procedure is potentially noisy because the answer for the sampled question could be non-zero also for the target image. From a manual inspection of 200 such questions, we found this to happen infrequently – about 4.5%.

Our extraction method covers various answer candidates such as compound nouns, noun phrases,

[1] https://spacy.io/

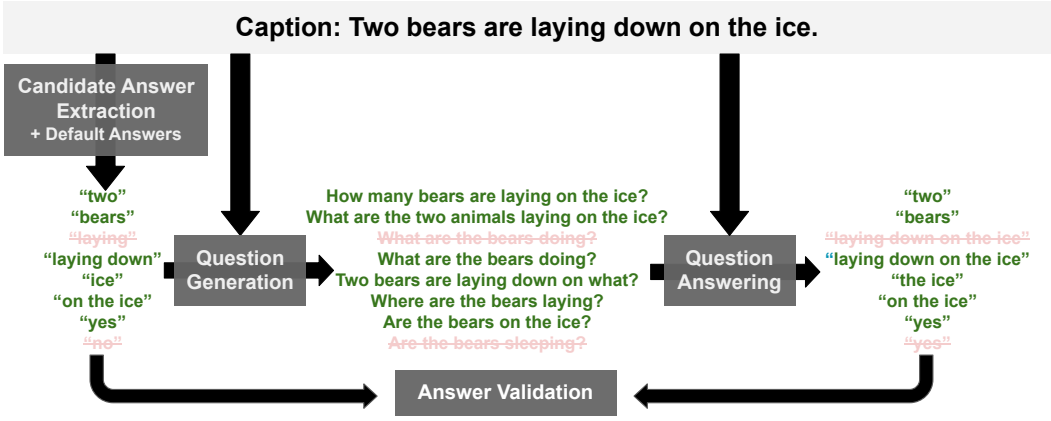

Figure 2: **V**isual **Q**uestion Generation with **Q**uestion **A**nswering validation (VQ^{2}A) has three main stages: Candidate Answer Extraction (Section 3.1), Question Generation (Section 3.2), and Question-Answering Filtering (Question Answering + Answer Validation, Section 3.3).

Candidate Answer	Noun Phrase	POS	Parse Tree	Boolean
'*two*'	V	V		
'*bears*'		V		
'*two bears*'	V		V	
'*laying*'		V		
'*laying down*'		V		
'*ice*'		V		
'*the ice*'	V			
'*on the ice*'			V	
'*no*'				V
'*yes*'				V

Table 1: Answer candidates extracted from the sentence "*two bears are laying down on the ice*" and the mechanism used to extract them.

named entities, boolean answers, cardinal and ordinal numbers, verbs and their compounds, (multi-word) adjectives and prepositional phrases. Table 1 provides an example of candidate answers of various types and the mechanism used to extract them.

3.2 Question Generation

Our question generation model, $\mathrm{q} = QG(\mathrm{a}, \mathrm{cap})$, takes as input a caption, cap, and a candidate answer span within it, a, and generates a question q, whose answer given the input caption is the input answer span. Importantly, the answer a does not need to appear verbatim in the caption, enabling the generation of questions for answer types like boolean and zero counts (see Section 3.1).

Given the advances in neural text generation, including models like T5 (Raffel et al., 2020), we choose to use a neural generation model as QG. Concretely, we use a T5-XXL model and further fine-tune it on SQuAD1.1 (Rajpurkar et al., 2016) for question generation. We take the top-scoring generated question for each caption-answer input. We note that our QG model is trained on a question answering dataset that is not caption-specific, and therefore is not optimized for caption inputs. From manual inspection of hundreds of generated questions, our QG model copes well with captions as input; see examples in Table 2 and Section 3.5.

3.3 Question-Answer Filtering

Generative models may hallucinate, that is, generate content that is inconsistent with its input source (Alberti et al., 2019; Honovich et al., 2021). To mitigate this, we follow (Alberti et al., 2019) and apply round-trip consistency by answering the generated question on the caption text with a question answering model. If the answer does not match the answer candidate offered as input to the question generation model, the generated question is discarded.

We use the token-level F1 score (Wang et al., 2020) to determine if the candidate answer and the QA model's answer is a match; If the score is above a threshold (manually set to 0.54, exemplified in Table 2), the pair is a match. For question answering, we use a T5-XXL model and further fine-tune it on SQuAD2.0 (Rajpurkar et al., 2018) and Natural Questions (Kwiatkowski et al., 2019).

3.4 Sources of Image/Caption Data

To gain insights on VQ^{2}A potential performance, we generate VQA triplets with VQ^{2}A from two sources of image captions: MSCOCO Captions (COCO-CAP) (Chen et al., 2015) and Conceptual Captions (CC3M) (Sharma et al., 2018). COCO-CAP captions contains 123,287 images from the COCO dataset (Lin et al., 2014), each with 5 *gold* captions manually created by raters with careful guidelines. CC3M contains 3.32M images

Candidate Answer	Generated Question	Validated Answer	Match Score & Result
'*two*'	'*How many bears are laying on the ice?*'	'*two*'	1.0 (Pass)
'*bears*'	'*What are the two animals laying on the ice?*'	'*bears*'	1.0 (Pass)
'*two bears*'	'*How many bears are laying on the ice?*'	'*two*'	1.0 (Pass)
'*laying*'	'*What are the bears doing?*'	'*laying down on the ice*'	0.4 (Fail)
'*laying down*'	'*What are the bears doing?*'	'*laying down on the ice*'	0.7 (Pass)
'*ice*'	'*Two bears are laying down on what?*'	'*the ice*'	1.0 (Pass)
'*the ice*'	'*Where are the bears laying?*'	'*on the ice*'	0.7 (Pass)
'*on the ice*'	'*Where are the bears laying?*'	'*on the ice*'	1.0 (Pass)
'*no*'	'*Are the bears sleeping?*'	'*yes*'	0.0 (Fail)
'*yes*'	'*Are the bears on the ice?*'	'*yes*'	1.0 (Pass)
'*zero*'	'*How many people are sitting down?*'	-	Pass by definition

Table 2: Question/answer pairs generated from the sentence "*two bears are laying down on the ice*" and the filtering decision. For answer candidate '*zero*', no validation is performed .

Dataset	Image		VQA examples	
	train	dev	train	dev
VQ^{2}A COCO	114.9K	8.4K	3.50M	257.5K
VQ^{2}A CC3M	3.32M	15.8K	13.29M	61.2K
COCOQA	64.5K	4.7K	108.7K	38.6K
VQA2.0	114.9K	8.4K	582K	65.1K
GQA	82.4K	0.4K	1.08M	12.6K
OKVQA	9K	5K	8.3K	4.7K

Table 3: **Sizes** of our generated VQ^{2}A data (top two rows) and VQA datasets used in our experiments.

automatically-collected from the web, each with one associated alt-text which we treat as a *silver* caption.

These datasets are quite different. Both the amount and the domain of CC3M images are larger and its captions look more plausible for capturing a larger set of object/attribute/action annotations. On the other hand, COCO-CAP's captions are cleaner and represent image content more adequately (see also Section 3.5). Thus, using COCO-CAP would show the potential of training a VQA model using VQ^{2}A in a "cleaner" zero-shot setup, where captions are human-curated. Using CC3M would indicate the potential of training on noisy web image–alt-text pairs, where scaling up to billions of examples is possible.

To quantify the impact of our method, we focus on VQA classification for the VQA2.0 (Goyal et al., 2017), GQA (Hudson and Manning, 2019), and OKVQA (Marino et al., 2019) benchmarks (see Section 4.2). We thus restrict our classifier to top 5,971 answers that are part of a unified answer vocabulary from these benchmarks (Appendix B.1). To this end, we remove triplets whose answers are not in the target answer vocabulary, and leave the study of using all generated triplets to future work. We then split our datasets into train/dev sets. In particular, since the images in VQA2.0 are taken from COCO, we split the COCO dataset based on the standard VQA2.0 train/dev splits of *train2014 and minival2014 (Jiang et al., 2018)[2]. For the CC3M dataset, we use the default CC3M train/dev splits (Sharma et al., 2018). For each unique image-question pair in the dev split, we construct an answer target of size 10, following VQA2.0, by reducing or expanding the set of seed answers that occur for this image-question pair. Additional details are in Appendix B.1.

Table 3 depicts the size of the induced datasets, named VQ^{2}A-COCO and VQ^{2}A-CC3M, as well as the VQA datasets used in our experiments.

3.5 Quality Analysis

To measure the quality of the generated datasets, we sampled 800 examples from each of the VQ^{2}A-COCO and VQ^{2}A-CC3M datasets. The sample was split between four authors, who assessed whether the answer to the question in an example is justified based on the example's image. For each dataset, 50 examples were rated by all raters, resulting in a free-margin Kappa (Randolph, 2005) of 0.71 for VQ^{2}A-COCO and 0.59 for VQ^{2}A-CC3M, corresponding to high inter-rater agreement. The measured percentage of valid triplets is 87.3% for VQ^{2}A-COCO and 66.0% for VQ^{2}A-CC3M. This shows the difference between the high-quality captions of COCO-CAP and the noisier web-based ones of CC3M.

Fig. 3 demonstrates the diversity of questions generates in the VQ^{2}A datasets. One can see that a significant amount of questions generated by VQ^{2}A for the shared VQA2.0/COCO image do not appear in VQA2.0. Additional analysis and examples are in Appendix A.

4 Visual Question Answering (VQA)

To assess the effectiveness of our automatic generation of VQA annotations, we perform extrinsic

[2]With the exception of OKVQA in which we split into train2014/val2014 to avoid using test images during training.

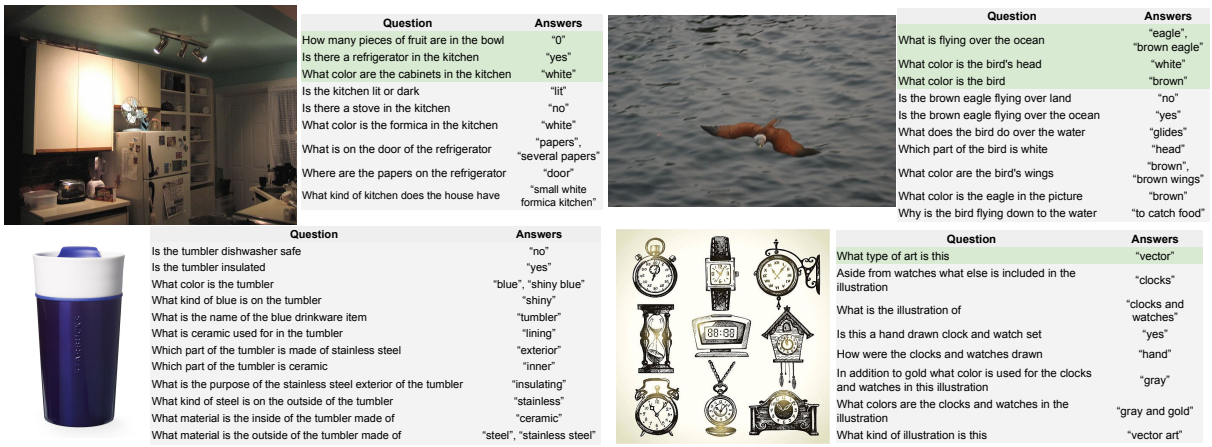

Figure 3: **Examples** from VQ^{2}A COCO (top) and VQ^{2}A CC3M (bottom). Questions with the green background are also present in VQA2.0.

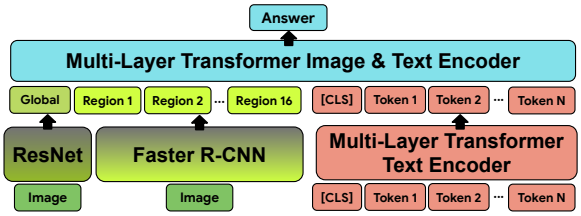

Figure 4: **VQA model** used in our experiments. The text encoder is initialized from a T5-base checkpoint, while the image-text encoder is initialized from scratch. The parameters of ResNet and Faster R-CNN are frozen during VQA training.

evaluations of the generated data by measuring its impact on a variety of established VQA benchmarks. We first describe the model, followed by the experimental setup and the results.

4.1 VQA Formulation and Model

Following the literature, we treat VQA as a classification task, i.e., vocab-based VQA. In particular, we treat our target answers as labels, where a label could be multi-token (e.g., "Christmas tree", "black and white", "play tennis"). We define our set of labels based on top answers in the training set of downstream VQA datasets, which allows for a fair comparison with most work in the VQA literature since Antol et al. (2015).

Since our work explores the impact of automatically-generated training data, we fix the VQA model architecture across all experimental conditions. Our model fuses the input image and question (Fig. 4). On the image side, we take global image features from ResNet-152 (He et al., 2016) pre-trained on ImageNet (Russakovsky et al., 2015) plus 16 region-of-interest image features from Faster R-CNN (Ren et al., 2015b) pre-trained on Visual Genome (Krishna et al., 2017), following the bottom-up-features paradigm (Anderson et al., 2018). On the question side, we use the encoder of a pre-trained T5-base checkpoint (Raffel et al., 2020). Given the image features and the output token embeddings of the question encoder, a Transformer (Vaswani et al., 2017) fuses the multi-modal intermediate representation and classifies it into the predefined answer space. We train the (randomly-initialized) fusing encoder and the text encoder end-to-end using standard cross-entropy loss. The parameters of both ResNet and Faster R-CNN are frozen during training. Additional details are given in Appendix B.2.

4.2 Experimental Setup

We consider three VQA benchmarks: VQA2.0 (Goyal et al., 2017), GQA (Hudson and Manning, 2019), and OKVQA (Marino et al., 2019). These datasets have their own characteristics and thus test different capability of VQA models. For instance, GQA puts emphasis on reasoning and OKVQA on external knowledge, whereas VQA2.0 is more general; VQA2.0 and GQA are order-of-magnitude larger than OKVQA; GQA is generated using a question engine while VQA2.0 and OKVQA are human-annotated.

For training and evaluating on VQA2.0, we use the standard train/dev splits *train2014 and minival2014 (Jiang et al., 2018). For GQA, we use the balanced v1.2 and combine the train and val splits for training and use the testdev split for evaluation, following the official guideline[3] and (Tan and Bansal, 2019). For OKVQA, we use the train/val splits for training/evaluation. Table 3 summarizes the sizes of the different datasets.

Evaluation Settings and Baselines. The main

[3] https://cs.stanford.edu/people/dorarad/gqa/evaluate.html

goal of our experiments is to explore the utility of our VQ^{2}A data for transfer learning, as training or evaluation data.

Our main focus in this paper is on zero-shot evaluation. Still, fine-tuning would provide additional insight on using our induced data for pre-training. Therefore, following (Banerjee et al., 2021), we train VQA models on the generated VQ^{2}A data and then evaluate them in two settings: (i) **zero-shot** evaluation, in which we evaluate our models as-is on the dev split of VQA2.0, GQA, or OKVQA; and (ii) **fully-supervised** fine-tuning evaluation, in which we further fine-tune our models on the training split of VQ^{2}A, GQA, or OKVQA before evaluating them. When training on VQ^{2}A data, we explore training on VQ^{2}A-COCO only, VQ^{2}A-CC3M only, and a two-stage training VQ^{2}A-CC3M followed by VQ^{2}A-COCO (VQ^{2}A CC3M $\rightarrow$ COCO).

Our baselines, which do not use VQ^{2}A data, include (i) our VQA model trained on template-based question generation data COCOQA[4] (Ren et al., 2015a), (ii) state-of-the-art zero-shot WeaQA (Banerjee et al., 2021) and its fully-supervised variants, and (iii) our VQA model trained supervisely on each of the target benchmarks' training data.

Metrics. To be compatible with prior work, on VQA2.0 and OKVQA we measure the standard *VQA Accuracy*. It is the average score over 9 subsets of the ground-truth 10 answers[5], where each score is: $min(\frac{\#answer\ occurrences}{3}, 1)$. On GQA, we measure *Top-1 Accuracy* against the single ground-truth answer.

5 Results

We report several sets of experimental results that shed light both on the accuracy and on the robustness of VQA models trained on VQ^{2}A data in this section, with additional results, analysis and ablation studies in Appendix C.

5.1 Zero-Shot Setting

Table 4 summarizes the outcomes of our VQA experiments on various benchmarks. Our main result is that the VQ^{2}A models achieve new state-of-the-art results in the zero-shot transfer learning setting. The improvement in performance is large: to the best of our knowledge, previous state-of-the-art zero-shot accuracy was 46.8% on VQA2.0 and 33.7% on GQA by WeaQA (Banerjee et al., 2021), which also induces their training VQA data from COCO Captions. Our VQ^{2}A-COCO model reaches 60.0% on VQA2.0 and 51.3% on GQA, an absolute improvement of +13.2% and +17.6%, respectively. Even higher accuracy for the zero-shot setting – 61.1% (VQA2.0) and 52.1% (GQA) – is reached with the VQ^{2}A CC3M $\rightarrow$ COCO model (trained first on the CC3M-derived data and then fine-tuned on the COCO-derived data), establishing new state-of-the-art results.

Approach	Evaluation Benchmark		
	VQA2.0	GQA	OKVQA
Zero-shot			
VQ^{2}A COCO, nouns only	10.5	-	-
COCOQA	11.7	4.4	6.3
WeaQA ZSL	46.8	33.7	-
VQ^{2}A COCO	60.0	51.3	18.0
VQ^{2}A CC3M	56.5	49.9	19.1
VQ^{2}A CC3M $\rightarrow$ COCO	61.1	52.1	19.7
VQ^{2}A CC3M +D	57.9	50.0	19.8
Fully-supervised			
WeaQA FSL	65.3	55.2	-
w/o VQ^{2}A data	68.8	61.8	22.1
w. VQ^{2}A COCO	71.6	63.3	36.0
w. VQ^{2}A CC3M	71.3	63.4	39.0
w. VQ^{2}A CC3M $\rightarrow$ COCO	71.4	64.0	39.3
Human performance	82.4†	89.3‡	82.8†

† from the inter-annotator agreement of ground-truth answers.
‡ from (Hudson and Manning, 2019).

Table 4: VQ^{2}A **as training data.** Accuracy in zero-shot and fully-supervised settings. All results use our architecture, except *WeaQA ZSL* and *WeaQA FSL*, which are the zero-shot (ZSL + Patches + Encoder) and fully-supervised (FSL + Patches + Encoder) models in (Banerjee et al., 2021), respectively. +D stands for recovered raw CC3M alt-texts with digits.

Training the same model architecture on the manually-constructed VQA2.0 and GQA training sets in a fully-supervised manner achieves 68.8% and 61.8% accuracy, respectively. Hence, our results significantly close the performance gap between automatically-generated and manually-constructed training sources, indicating that the VQ^{2}A method may reduce the need for human curated VQA training examples.

The captions for COCO images are carefully annotated to be of high-quality (Chen et al., 2015). Additionally, the VQA2.0 images are taken from COCO. To test the robustness of VQ^{2}A, we also evaluate a VQ^{2}A-CC3M model. While CC3M contains more image–alt-text pairs than COCO (see Table 3), the images are from a different distribution and the text annotations are noisier and may represent a larger spectrum of discourse intents

[4]Train/dev based on the standard VQA2.0 train/dev splits.
[5]5 targets in OKVQA, replicated twice (Marino et al., 2019).

(Alikhani et al., 2020). In spite of these differences, the gap between COCO-based and CC3M-based VQ^{2}A models is not large, 60.0% vs 56.5% on VQA2.0 and 51.3% vs. 49.9% on GQA. This result strengthens our previous observation, in that it does not seem to be crucial that the starting captions are manually-annotated; it appears that "silver" annotations such as the ones provided by CC3M are competitive in zero-shot VQA performance.

To cover the types of answers present in VQA benchmarks, there is a need for thorough extraction of various answer/question types (Section 3). The QACE model (Lee et al., 2021), for example, focuses only on noun-phrases as answer types. By analyzing the VQA2 devset, we find that only 32% of its answers are nouns. As such, it makes sense that, when limiting to only this answer type, the *VQA Accuracy* of VQ^{2}A-COCO is 10.5%, compared to the 60% achieved with a full coverage. As another example, our model trained COCOQA (Ren et al., 2015a), which focuses on a few answer types and one-word answers, barely surpasses the accuracy of our *COCO, nouns only* baseline. For similar reasons, we want to be able to generate '*how many*' questions from the CC3M data, even though the published annotations have been stripped of digits and numerals. To solve this problem, we recover the original captions from the CC3M URLs, generate questions of the type '*how many*', and train an additional VQ^{2}A-CC3M +D model. The results in Table 4 show a small but consistent improvement over vanilla VQ^{2}A-CC3M, further closing the gap between VQ^{2}A models using curated "gold" captions and noisier "silver" captions.

To gain further insights, we provide a breakdown of *VQA Accuracy* per VQA2.0 question types in Table 5. Boolean questions are the easiest and all models perform well on them. More challenging question types are '*How many?*' and '*What is*'. One reason could be the validity of various answers, like "*several*" for counts. '*What time?*' is the most difficult, probably due to lack of such information in captions.

Finally, we provide zero-shot results on the more difficult OKVQA benchmark. In this setting, a supervised model reaches 22.1% accuracy, while VQ^{2}A models in zero-shot setting achieve close to that – 18.0% with COCO and 19.1% with CC3M, while their combination reaches 19.7%, -2.4% shy of the supervised level. This result also supports the conclusion that creating training data with the VQ^{2}A method is a good replacement for small-scale supervised training data.

Question Prefix	VQA2.0 Supervised	VQ^{2}A-COCO Zero-shot	VQ^{2}A-CC3M Zero-shot
Boolean	96.3	93.2	94.2
'*What color*'	69.2	64.8	56.8
'*What kind/is*'	52.6	36.9	32.1
'*How many*'	49.3	29.4	19.5
'*Where are/is*'	38.0	30.0	25.3
'*What does*'	33.0	24.1	20.3
'*What time*'	23.6	11.9	12.7

Table 5: Aggregated average Accuracy on VQA2.0 for the most common question types.

5.2 Fully-Supervised Setting

Another aspect of the VQ^{2}A method that we want to evaluate is whether it produces training data that is similar with the human-annotated data, or it complements it. To this end, we perform experiments in which we first train a model using the VQ^{2}A data, and then fine-tune it in a supervised manner using the human-annotated training data.

The results, in the *Fully-supervised* part of Table 4, tell two stories. For VQA2.0 and GQA, there is a small yet consistent improvement of the fine-tuned models on top of a model trained directly on the supervised data in each benchmark (labeled *w/o* VQ^{2}A). This indicates that, at least for these two benchmarks, there is a high overlap in the nature of the signal between the human-annotated data and the VQ^{2}A data.

The results on OKVQA show a different trend. Here, training first with VQ^{2}A boosts performance by +17.2% compared to supervised training without VQ^{2}A (22.1% $\rightarrow$ 39.3%). The small scale of the OKVQA training set (Table 3) certainly contributes to this effect, but it also points to another aspect: question-answer pairs that subsume world knowledge can only be made available at-scale to models by means that are not bottlenecked by human-annotation processes.

5.3 Robustness of Existing VQA Training Sets

So far we have assessed the capability of models trained on VQ^{2}A data. As a complementary study, we use 500 manually-validated random samples (see Section 3.5) from the dev part of each VQ^{2}A dataset to assess VQA robustness for various training setups. We use the *VQA Accuracy* metric for the VQ^{2}A datasets (10 target answers, see Section 3.4), and *Top-1 Accuracy* on COCOQA (one target answer).

Table 6 shows the results. The fully-supervised

Training data	Evaluation Benchmark (Acc %)			
	COCO-QA	VQA2.0	VQ^{2}A COCO	VQ^{2}A CC3M
COCOQA	70.3	11.7	13.2	5.8
VQA2.0	35.9	68.8	44.4	41.6
VQ^{2}A COCO	**55.9**	**60.0**	72.6	**56.8**
VQ^{2}A CC3M	42.1	56.5	**65.6**	76.4

Table 6: **Manually-validated** VQ^{2}A **data for robustness evaluation**: Accuracy of training on "row" and tested on "column"; diagonal (gray) numbers denote supervised setting, non-diagonal numbers denote zero-shot cross-dataset setting. Best zero-shot is in **bold**.

models (diagonal, similar training and test distributions) achieve in-domain Accuracy around 70%, with VQ^{2}A CC3M achieving slightly higher 76.4% Accuracy. When tested on out-of-domain (non-diagonal), however, each model poses performance degradation at different degrees. First, the model based on template-generated COCOQA does not generalize at all. Second, the VQA2.0 model sees significant accuracy drops, even on the COCO (44.4%) and COCOQA (35.9%), which share a similar image domain with VQA2.0. This result provides another evidence that progress made on the VQA2.0 benchmark may not reflect progress on the VQA task in full (Chao et al., 2018a; Bras et al., 2020).

In contrast, both VQ^{2}A COCO and VQ^{2}A CC3M perform robustly with more modest performance drops. For instance, on COCOQA, VQ^{2}A CC3M achieves even better performance than VQA2.0 (42.1% vs. 35.9%) despite being tested on out-of-domain images. This suggests that the VQ^{2}A training data possesses a higher degree of question variations, provides better answer coverage, and exhibits less biases than the manually-curated VQA2.0 training data, at least enough to address these different benchmarks.

6 Considerations and Limitations

Automatic data generation is prone to erroneous outputs. In VQ^{2}A these may include hallucinations of the generative model, incorrect negative sampling, and bad answer span extraction. In addition, the image captions may contain details not in the image, e.g. additional details only aware to the photo taker or personal opinions, or information that is inconsistent with the image due to human mistakes and biases. We address some of these issues automatically, filtering bad generations via question answering round-trip validation. In addition, the classification task itself curbs the effects of such errors through the use of a fixed answer vocabulary. Yet, for automatic generation to be more robust, additional methods to narrow down mistakes or mismatches need to be developed.

The resulting VQA model incorporates and may reinforce some of the biases and stereotypes present in the data. For instance, it may learn that answering questions such as “What is the gender of this person?” is a binary choice dictated by shallow cues, or that the answer to “For whom is this room decorated?” depends on stereotypical features present (or not) in the room depicted in the image. Mitigation strategies for such issues go beyond the scope of this paper, but we encourage the research community to consider addressing these issues as central for the successful deployment of this technology.

7 Conclusions

In this paper, we show that high-quality VQA training data can be automatically induced at scale from existing image-caption datasets. Our method, VQ^{2}A, annotates candidate answers using syntactic parsing of the captions and then derives questions for them using neural models for question generation and question answering verification. We demonstrate that VQA models trained only on such data exhibit high zero-shot performance with new state-of-the-art results on VQA2.0 and GQA. Additionally, we provide evidence for the brittleness of VQA systems built with human-annotated examples compared to the ones built with automatically-induced image-question-answer triplets using VQ^{2}A.

For future work, we plan to explore even larger automatically-curated image-text datasets, consisting of billions of examples. In addition, we want to test the applicability of VQ^{2}A to languages other than English, for which human-annotated VQA data is scarce.

Acknowledgments. We would like to thank Or Honovich, Hagai Taitelbaum and Roee Aharoni for their help with question generation, Sebastian Goodman for his help with the VQA infrastructure, Piyush Sharma for his help with the Conceptual Captions, Nassim Oufattole for his early exploration of question generation, Gal Elidan, Sasha Goldshtein, and Avinatan Hassidim for their useful feedback.

References

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *CVPR*.

Arjun Akula, Soravit Changpinyo, Boqing Gong, Piyush Sharma, Song-Chun Zhu, and Radu Soricut. 2021. CrossVQA: Scalably generating benchmarks for systematically testing vqa generalization. In *EMNLP*.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *ACL*.

Malihe Alikhani, Piyush Sharma, Shengjie Li, Radu Soricut, and Matthew Stone. 2020. Cross-modal coherence modeling for caption generation. In *ACL*.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual question answering. In *ICCV*.

Pratyay Banerjee, Tejas Gokhale, Yezhou Yang, and Chitta Baral. 2021. WeaQA: Weak supervision via captions for visual question answering. In *Findings of ACL-IJCNLP*.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *ICML*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*.

Wei-Lun Chao, Hexiang Hu, and Fei Sha. 2018a. Being negative but constructively: Lessons learnt from creating better visual question answering datasets. In *NAACL*.

Wei-Lun Chao, Hexiang Hu, and Fei Sha. 2018b. Cross-dataset adaptation for visual question answering. In *CVPR*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft COCO Captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: Learning UNiversal Image-TExt Representations. In *ECCV*.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In *ICML*.

Kaustubh D. Dhole and Christopher D. Manning. 2020. Syn-QG: Syntactic and shallow semantic rules for question generation. In *ACL*.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *ACL*.

Adam Fisch, Kenton Lee, Ming-Wei Chang, Jonathan H. Clark, and Regina Barzilay. 2020. CapWap: Captioning with a purpose. In *EMNLP*.

Manas Gaur, Kalpa Gunaratna, Vijay Srinivasan, and Hongxia Jin. 2021. Iseeq: Information seeking question generation using dynamic meta-information retrieval and knowledge graphs. *arXiv preprint arXiv:2112.07622*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*.

Mandy Guo, Qinlan Shen, Yinfei Yang, Heming Ge, Daniel Cer, Gustavo Abrego, Keith Stevens, Noah Constant, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Effective parallel corpus mining using bilingual sentence embeddings. In *WMT*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Michael Heilman and Noah A. Smith. 2009. Question generation via overgenerating transformations and ranking. Technical report, Carnegie Mellon University.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. Q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In *EMNLP*.

Drew A. Hudson and Christopher D. Manning. 2019. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*.

Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2018. Pythia v0.1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*.

Kushal Kafle and Christopher Kanan. 2017. An analysis of visual question answering algorithms. In *ICCV*.

Kushal Kafle, Mohammed Yousefhussien, and Christopher Kanan. 2017. Data augmentation for visual question answering. In *INLG*.

Jihyung Kil, Cheng Zhang, Dong Xuan, and Wei-Lun Chao. 2021. Discovering the unknown knowns: Turning implicit knowledge in the dataset into explicit training examples for visual question answering. In *EMNLP*.

Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. 2019. Information maximizing visual question generation. In *CVPR*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael Bernstein, and Li Fei-Fei. 2017. Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, , Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a benchmark for question answering research. *TACL*, 7:453–466.

Hwanhee Lee, Thomas Scialom, Seunghyun Yoon, Franck Dernoncourt, and Kyomin Jung. 2021. QACE: Asking questions to evaluate an image caption. In *Findings of EMNLP*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. VisualBERT: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*.

Yikang Li, Nan Duan, Bolei Zhou, Xiao Chu, Wanli Ouyang, Xiaogang Wang, and Ming Zhou. 2018. Visual question generation as dual task of visual question answering. In *CVPR*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2014. Microsoft COCO: Common objects in context. In *ECCV*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-task vision and language representation learning. In *CVPR*.

Chenyang Lyu, Lifeng Shang, Yvette Graham, Jennifer Foster, Xin Jiang, and Qun Liu. 2021. Improving unsupervised question answering via summarization-informed question generation. In *EMNLP*.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. OK-VQA: A visual question answering benchmark requiring external knowledge. In *CVPR*.

Yosi Mass, Boaz Carmeli, Haggai Roitman, and David Konopnicki. 2020. Unsupervised FAQ retrieval with question generation and BERT. In *ACL*.

Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. 2016. Generating natural questions about an image. In *ACL*.

Shashi Narayan, Gonçalo Simoes, Ji Ma, Hannah Craighead, and Ryan Mcdonald. 2020. QURIOUS: Question generation pretraining for text generation. *arXiv preprint arXiv:2004.11026*.

Preksha Nema, Akash Kumar Mohankumar, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2019. Let's ask again: Refine network for automatic question generation. In *EMNLP-IJCNLP*.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. In *EMNLP*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *ACL*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Justus J. Randolph. 2005. Free-marginal multirater kappa (multiraterκ_{free}): An alternative to Fleiss' fixed-marginal multirater kappa. *Joensuu Learning and Instruction Symposium*.

Mengye Ren, Ryan Kiros, and Richard Zemel. 2015a. Exploring models and data for image question answering. In *NIPS*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015b. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet large scale visual recognition challenge. *IJCV*, 115(3):211–252.

Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. 2019. Cycle-consistency for robust visual question answering. In *CVPR*.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual Captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *ICLR*.

Amanpreet Singh, Vedanuj Goswami, Vivek Natarajan, Yu Jiang, Xinlei Chen, Meet Shah, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2020. MMF: A multimodal framework for vision and language research. https://github.com/facebookresearch/mmf.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. VL-BERT: Pre-training of generic visual-linguistic representations. In *ICLR*.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *EMNLP-IJCNLP*.

Damien Teney and Anton van den Hengel. 2016. Zero-shot visual question answering. *arXiv preprint arXiv:1611.05546*.

Kohei Uehara, Antonio Tejero-de-Pablos, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Visual question generation for class acquisition of unknown objects. In *ECCV*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *ACL*.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2021. SimVLM: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*.

Jialin Wu, Zeyuan Hu, and Raymond J. Mooney. 2019. Generating question relevant captions to aid visual question answering. In *NeurIPS*.

Xing Xu, Tan Wang, Yang Yang, Alan Hanjalic, and Heng Tao Shen. 2021. Radial graph convolutional network for visual question generation. *IEEE Transactions on Neural Networks and Learning Systems*, 32(4):1654–1667.

Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. 2021a. Just ask: Learning to answer questions from millions of narrated videos. In *ICCV*.

Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. 2018. Visual curiosity: Learning to ask questions to learn visual recognition. In *CoRL*.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. 2021b. An empirical study of GPT-3 for few-shot knowledge-based VQA. *arXiv preprint arXiv:2109.05014*.

Desen Yuan. 2021. Language bias in visual question answering: A survey and taxonomy. *arXiv preprint arXiv:2111.08531*.

Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. 2021. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. VinVL: Revisiting visual representations in vision-language models. In *CVPR*.

Shijie Zhang, Lizhen Qu, Shaodi You, Zhenglu Yang, and Jiawan Zhang. 2017. Automatic generation of grounded visual questions. In *IJCAI*.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and VQA. In *AAAI*.

A Additional Examples and Analysis of Generated Data

Fig. 5 provides additional examples of VQ^{2}A COCO and CC3M generated VQA triplets, showing the diversity compared to what can be found in VQA2.0.

Table 7 presents the top question prefixes and their distribution in the VQA2.0 and VQ^{2}A-based dev sets, showing significant differences between datasets. Many questions in VQA2.0 are of boolean answer type, e.g. '*is the*', '*is there*' and '*does the*', summing to 29.2%. In addition, ('*how many*') questions are frequent, 11%. Finally, questions for the color attribute are standing out with 9%. On the other hand, COCO and CC3M questions are more explanatory in nature, with the majority of questions (45.5% in COCO, 43.9% in CC3M) of the form '*what is/are/do/does/type*'. Another type that is more prominent in COCO and CC3M are '*where is/are*' questions, which are more than twice frequent compared to VQA2.0.

Another difference between the manually curated VQA2.0 dataset and the VQ^{2}A automatically generated datasets is question and answer word length distribution (Fig. 6 and 7). The questions in VQ^{2}A-CC3M and VQ^{2}A-COCO have an average word length of 8.3 and 7.8 respectively, while the average VQA2.0 is 6.3. Inspecting the generated questions, we noticed that QG model tends to quote parts of the caption, extending the question length. The average answer word length in VQ^{2}A-CC3M and VQ^{2}A-COCO is 1.76 and 1.85 words respectively, while in VQA2.0 it is 1.1. While all answers tend to be short, the VQ^{2}A-induced datasets have more "detailed" answers of length 2-3 words.

Fig. 8 offers a more visual view of the differences between question type distribution presented in Table 7.

Table 8 depicts the percentage of questions of each type (prefix) that were retained (not filtered out) when applying the question answer validation phase of VQ^{2}A (Section 3.3).

B Implementation Details

B.1 Details on Data Processing

Our default question and answer preprocessor is based on (Jiang et al., 2018; Singh et al., 2020)[6], with the exception of GQA which we use [7]. The unified answer vocabulary used in our experiments is the union of top answers from existing COCO-based VQA benchmarks: VQA2.0 (3,128, minimum answer frequency=9), GQA (1,843, all), OKVQA (2,000, top), and Visual7W (3,140, minimum answer frequency=3) of total size 5,971

For each image-unique question pair generated by our VQ^{2}A approach, we reduce or expand a list of possibly different candidate answers based on the list length, such that we eventually have a target list of answers of size 10. In particular, we first sort the answers based on their lengths ("dog" before "black dog"), and select up to top-10 answers. If the list legnth is less than 10, we replicate each of the top answers one-by-one until we have the list of size 10, similar to the process in OKVQA(Marino et al., 2019). This is to ensure that we can adopt *VQA Accuracy* to make the performance comparison.

B.2 Details on Training and Evaluating Visual Question Answering

Our code for the VQA model is based on the Flaxformer framework[8]. Both the text encoder and the multi-modal encoder have 6 blocks of Transformers, each of which consists of self-attention and a feed-forward network. We use 12 heads of inner dimension of 64, the embedding dimension of 768, and the MLP dimension of 2048. During training, we use Adafactor (Shazeer and Stern, 2018), with an initial learning rate of 0.0025, a linear warm-up step of 5K for (pre-)training and 1K for fine-tuning, and an "inverse square root" learning rate schedule $\frac{1}{\sqrt{\max(n,k)}}$, where n is the current training iteration and k is the number of warm-up steps. We use a dropout rate of 0.0. We train each of the models with data parallelism using 16 Cloud TPU Pods[9], each with a batch size of 256, unless otherwise stated.

The default numbers of training steps during training and fine-tuning are 100K and 30K, respectively. The exceptions are OKVQA (30K/15K) and VQ^{2}A CC3M (150K/NA). In addition, in the two-stage training where we fine-tune a VQ^{2}A-CC3M model with VQ^{2}A COCO, we also use 100K steps. Each single training run on average took fewer than 10 hours, including the time used to evaluate

[6] https://github.com/facebookresearch/mmf/blob/main/mmf/datasets/processors/processors.py

[7] https://github.com/stanfordnlp/mac-network/blob/gqa/preprocess.py

[8] https://github.com/google/flaxformer

[9] https://cloud.google.com/tpu

Figure 5: Additional examples from VQ^{2}A COCO (top) and VQ^{2}A CC3M (bottom). Questions with the green background are present in VQA2.0.

Question Prefix	VQA2.0 %	VQ^{2}A-COCO %	VQ^{2}A-CC3M %	Question Example from VQ^{2}A-COCO
'*What is*'	0.140	0.288	0.217	'*What is the man swinging?*'
'*How many*'	0.110	0.022	0.005	'*How many people are standing in front of a tv?*'
'*Is the*'	0.098	0.084	0.053	'*Is the baby wearing a Santa hat?*'
'*What color*'	0.090	0.022	0.018	'*What color is the man's hair?*'
'*Is this*'	0.082	0.008	0.015	'*Is this a safe way to fly?*'
'*Is there*'	0.037	0.011	0.022	'*Is there a pool in the backyard?*'
'*What kind*'	0.025	0.049	0.078	'*What kind of truck is the yellow one?*'
'*What are*'	0.024	0.049	0.022	'*What are the sheep and other animals roaming?*'
'*Are the*'	0.024	0.022	0.007	'*Are the apples on the cutting board green?*'
'*Are there*'	0.020	0.002	0.004	'*Are there any exceptions to this rule?*'
'*Where is*'	0.019	0.071	0.034	'*Where is the tennis player pictured?*'
'*What type*'	0.018	0.006	0.022	'*What type of picture is this?*'
'*Is it*'	0.017	0.001	0.005	'*Is it possible to eat a whole pizza?*'
'*Does the*'	0.014	0.007	0.007	'*Does the adult giraffe have any young?*'
'*What does*'	0.011	0.015	0.038	'*What does a giraffe do with its long neck?*'
'*Where are*'	0.006	0.032	0.014	'*Where are the skateboarders in the photo?*'
'*Who is*'	0.005	0.054	0.020	'*Who is in the photo?*'
'*What do*'	0.002	0.003	0.018	'*What do the father and son ride?*'
'*What was*'	0.000	0.009	0.023	'*What was the woman looking at?*'
'*What did*'	0.000	0.001	0.021	'*What did the cat lay inside of?*'

Table 7: Most popular question prefix distribution on valid questions whose answers are in the 6k target vocabulary.

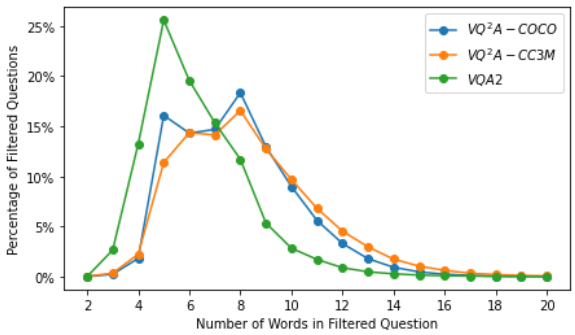

Figure 6: Question length distributions per dataset.

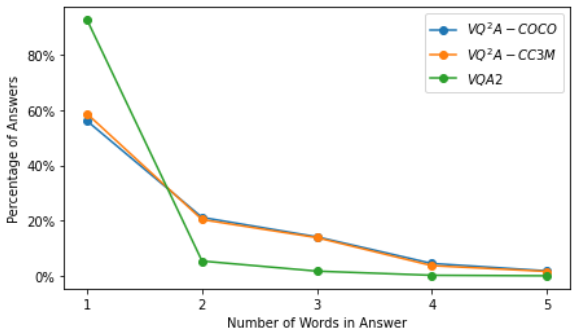

Figure 7: Answer length distributions per dataset.

a checkpoint — every 1K iterations. For instance, training on VQA2.0 took approximately 7 hours, VQ^{2}A COCO 13 hours, VQ^{2}A CC3M 10 hours. Note that VQ^{2}A COCO has larger evaluation set than other datasets, hence taking longer time to to train then VQ^{2}A CC3M.

The hyperparameters for Transformers are selected to be consistent with a T5-base checkpoint, which has 220 million parameters (Raffel et al., 2020) (except that now we have 2 encoders rather

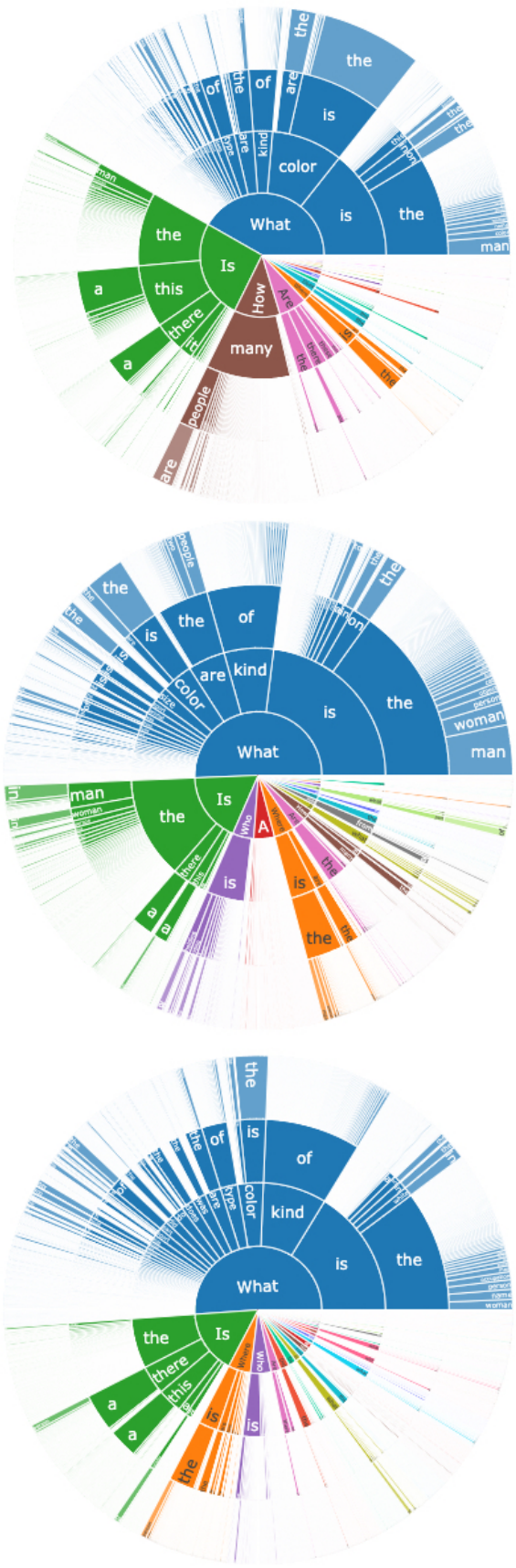

Figure 8: VQA2.0 (top), VQ^{2}A-COCO (middle), VQ^{2}A-CC3M (bottom) sunburst plots of question prefixes.

Question Prefix	VQ^{2}A-COCO Filter Pass Ratio	VQ^{2}A-CC3M Filter Pass Ratio
'*What is*'	0.73	0.65
'*Is the*'	0.64	0.39
'*What kind*'	0.84	0.80
'*How many*'	0.83	0.51
'*What color*'	0.92	0.90
'*Where is*'	0.79	0.79
'*Is this*'	0.83	0.62
'*What are*'	0.75	0.71
'*Who is*'	0.85	0.79
'*Is there*'	0.73	0.47
'*What does*'	0.75	0.67
'*Are the*'	0.58	0.32
'*Where are*'	0.80	0.81
'*What type*'	0.84	0.81
'*What was*'	0.72	0.67
'*Does the*'	0.60	0.43
'*Are there*'	0.80	0.62
'*What do*'	0.76	0.72
'*What did*'	0.69	0.64
'*Is it*'	0.62	0.59

Table 8: Question filtering stats.

than an encoder and a decoder). We initially tuned the initial learning rate (0.0125, 0.075, 0025, 0.00125, 0.00075) and the dropout rate (0.0, 0.1, 0.2) on a fully-supervised model on VQA2.0 baseline using *VQA Accuracy* and observed that 0.0025 and 0.0 work robustly across our experiments but we did not extensively tuned them in all of our experiments.

We implement *VQA Accuracy* ourselves based on the official challenge page for VQA2.0[10].

C Additional Results

Table 9 offers the Accuracy of the supervised VQA2.0 model, as well as of the zero-shot VQ^{2}A models, on the VQA2.0 devset, split by most common question prefixes. The Table is sorted by the supervised model's Accuracy. It shows a several performance differences, first between all types of boolean questions, which all have high precision on all models, vs. other types, which show not only lower performance for all models, but also more significant performance drop between the supervised and zreo-shot models.

Table 10 shows the zero-shot performance of models when using all of the VQ^{2}A dev sets, not only the manually validated sample, for which Table 6 reports results. What we see is that the difference in performance on the whole VQ^{2}A dev sets (Table 10) is similar in magnitude to that of the manually validated dev samples (Table 6), and most

[10] https://visualqa.org/evaluation.html

Question Prefix	VQA2.0 Supervised	VQ^{2}A-COCO Zero-shot	VQ^{2}A-CC3M Zero-shot
'*is there*'	98.6	98.1	98.2
'*are there*'	98.0	97.1	97.2
'*does this*'	98.0	95.1	95.8
'*are they*'	96.9	95.0	95.3
'*does the*'	96.4	95.2	95.9
'*is it*'	96.3	91.4	92.7
'*is this*'	96.1	91.2	92.8
'*are the*'	95.6	92.1	93.1
'*is the*'	95.3	91.7	92.9
'*are these*'	95.1	90.7	92.2
'*what color*'	69.2	64.8	56.8
'*what kind*'	56.3	35.8	31.4
'*what type*'	54.4	32.3	30.8
'*what are*'	51.3	40.2	33.9
'*how many*'	49.3	29.4	19.5
'*what is*'	48.5	39.4	32.2
'*where are*'	40.9	33.9	27.6
'*where is*'	35.1	26.0	23.0
'*what does*'	33.0	24.1	20.3
'*what time*'	23.6	11.9	12.7

Table 9: Average accuracy (%) on VQA2.0 for the most common question prefixes.

Training data	Evaluation Benchmark			
	COCO-QA	VQA2.0	VQ^{2}A COCO	VQ^{2}A CC3M
COCOQA	70.3	11.7	11.5	3.7
VQA2.0	35.9	68.8	41.1	33.3
VQ^{2}A COCO	**55.9**	**60.0**	71.2	**49.3**
VQ^{2}A CC3M	42.1	56.5	**60.3**	69.5

Table 10: VQ^{2}A **as evaluation data** for measuring robustness: VQA Accuracy when training on "row" and tested on "column"; diagonal (gray) numbers denote the supervised setting, non-diagonal numbers denote zero-shot cross-dataset setting. Best zero-shot is **bold**.

importantly, it keeps the order of models in terms of capabilities/performance. We therefore suggests that the utility of the VQ^{2}A approach could go beyond training; it can be used as an automatic test-bed for VQA robustness, if not for absolute figures, for ranking models for robustness zero-shot capabilities.

Table 11 shows the effect of candidate answer types on the VQA2.0 performance. We train our model on VQ^{2}A COCO or VQ^{2}A CC3M subsets with questions with (i) noun answers, (ii) yes/no answers, (iii) answers containing color-related tokens based on a list of common colors from Wikipedia, and (iv) answers containing digits from 0 to 100. We then evaluate models trained on these subsets on VQA2.0 using *VQA Accuracy* and the normalized version (by the percentage of evaluation questions with corresponding answer types. This highlights the importance of incorporating diverse answer candidates in our datasets. We also observe that VQ^{2}A CC3M is on par with VQ^{2}A COCO on

Training data	*VQA Accuracy* on VQA2.0	
	Standard	Normalized
VQ^{2}A COCO	60.0	60.0
VQ^{2}A COCO nouns	10.5	32.5
VQ^{2}A COCO yes/no	38.4	94.3
VQ^{2}A COCO color	6.7	55.6
VQ^{2}A COCO number	3.9	25.4
VQ^{2}A CC3M	56.5	56.5
VQ^{2}A CC3M nouns	8.8	27.2
VQ^{2}A CC3M yes/no	38.4	94.4
VQ^{2}A CC3M color	6.0	49.5
VQ^{2}A CC3M number	3.4	22.1

Table 11: **Effect of candidate answer types** on the VQA2.0 performance.

Training data	*VQA Accuracy* on VQA2.0
VQ^{2}A COCO (100%)	60.0
VQ^{2}A COCO (50%)	58.5
VQ^{2}A COCO (20%)	56.7
VQ^{2}A COCO (10%)	55.4
VQ^{2}A CC3M (100%)	56.5
VQ^{2}A CC3M (50%)	55.8
VQ^{2}A CC3M (20%)	54.8
VQ^{2}A CC3M (10%)	53.8

Table 12: **Effect of dataset sizes** on the VQA2.0 performance.

yes/no-answer questions but are behind on nouns, color, and number, which we attribute to their lower degree image-text relevance, less mentioning of colors (due to the style of alt-texts vs. captions), and digit substitution.

Table 12 shows the effect of scale on the VQA2.0 performance. We randomly sampled 10%, 20%, and 50% of VQ^{2}A COCO or VQ^{2}A CC3M training data. We observe that the bigger the data, the higher the accuracy. However, the gain is diminishing. We identify improving the data generation process to achieve higher degree of diversity in the output as interesting future work.

Table 13 provides question-only baselines (no image features as input). Interestingly, the models trained on our generated VQ^{2}A data has similar answer distributions to those of existing VQA benchmarks. At the same time, this reveals the exploitation of the language bias, suggesting that additional research on bias mitigation is needed, both in terms of model and data (existing benchmarks and our datasets).

D Further Considerations

Information that names or uniquely identifies individual people or offensive content. COCO

Approach	Evaluation Benchmark		
	VQA2.0	GQA	OKVQA
Zero-shot			
questions VQ^{2}A COCO	48.9	44.4	11.4
questions VQ^{2}A CC3M	47.8	44.6	11.9
VQ^{2}A COCO	60.0	51.3	18.0
VQ^{2}A CC3M	56.5	49.9	19.1

Table 13: **Zero-shot question-only baselines** using VQ^{2}A as training data.

Captions are human-curated and cleaned while the approach to collection of CC3M upholds rigorous privacy and ethics standards such as the removal of offensive content and hypernymization. This significantly mitigates the risks that our VQ^{2}A datasets would contain such information.

Intended uses. Due to considerations and limitations as we mention in Section 6, COCO Captions, CC3M, and our induced VQ^{2}A are intended to be used for research-only purposes.